



OLEKSII
MAZURENKO
FULL-STACK-ENTWICKLER

EMAIL newiqa@gmail.com
TELEFON +380 63 588 3160
WEBSITE oleksiimazurenko.dev
GITHUB github.com/oleksiimazurenko
LINKEDIN linkedin.com/in/oleksii-mazurenko
TELEGRAM @oleksii_mazurenko
MODUS Remote

KENNTNISSE

Kernstack

Sprachen:
TypeScript, JavaScript
Markup & Styles:
HTML, CSS/SCSS
Kern:
React 19 · Next.js — App Router, RSC, SSR/SSG/ISR, Server Actions, Route Handlers, proxy (middleware)
Styling / UI:
TailwindCSS · shadcn / Base UI · Framer Motion · GSAP (einfache Animationen) · Lexical · dnd-kit
State & Daten:
Zustand · SWR · TanStack Query · Zod · Prisma · MongoDB (Atlas) und PostgreSQL auf CRUD-Niveau
Architektur:
modulare Struktur für den App Router · Datenschema-Design (Prisma)
SEO / i18n:
sitemaps, robots.txt, Meta-Tags, JSON-LD, Canonical, ARIA, semantisches Markup · next-intl · Lingui
Analytics:
Amplitude (Event-Instrumentierung) · eigenes System zur Event-Erfassung · Faro (Grafana) · oxlint · Vercel
KI-Entwicklung:
Claude Code täglich, Agenten mit Skills für Routine, MCP-Integrationen

Integrationen

Solidgate · Stripe · Better Auth · Firebase Auth · GrowthBook · PostHog · Strapi CMS · Payload CMS · AWS (Lambda, SST + OpenNext) · Hetzner · OVH

Außerdem gearbeitet mit

Strapi 4/5 + Plugins · Build-Skripte und Codegen · Playwright (visuelle Regression) · pnpm · Turborepo · CI/CD · Solid.js · Svelte

MEINE ARBEITSWEISE

1 · Erst die Aufgabe abschaffen, dann beschleunigen

- Agent übernimmt** — die Aufgabe kommt zu niemandem zurück: automatisches Code-Review, Analytics landen selbst in Slack.
- Auftraggeber übernimmt** — die Aufgabe erreicht mich nicht: der Anfragende erledigt sie mit KI (Outreach, Influencer-Plattform).
- Team übernimmt** — ein Plugin, über das das Team selbst ändert: CMS-Weiterleitungen für das SEO-Team.

2 · Wenn ich es selbst mache

- Wiederkehrendes Muster — dafür schreibe ich zuerst einen Skill.
- Plan-Modus vor dem Code: zuerst Umfang und Plan, dann die Umsetzung — selbst oder mit Agent, je nach Fehlerkosten.

3 · Ablauf

Dev → Build → Unit-Tests + Playwright (**Agent, ich lese**) → Staging (**mein Check**) → QA → PR (Agenten mit Skills je Projektzone) → Produktion → Endkontrolle

Grundlage: OOP · SOLID · DRY · KISS · YAGNI · CI/CD + eine strikte Schrittfolge, beschrieben unter: [oleksiimazurenko.dev/pipeline](#)

SPRACHEN

Ukrainisch — Muttersprache
Englisch — A2: Fachdokumentation lese ich, Sprechen übe ich täglich
Deutsch — Grundkenntnisse

FULL-STACK-ENTWICKLER

REACT / NEXT.JS

Full-Stack-Entwickler, 5+ Jahre mit React / Next.js; Serverseite im Rahmen von Next.js (Server Actions, Prisma). Ich **bewerte Aufgaben nach ihrem Geschäftswert** und verantworte meinen Bereich — von der Entscheidung bis zur Produktion. Im ersten Jahr bei Promova habe ich das Produkt selbst gebaut, danach kamen KI-Agenten dazu: Routine schreibe ich mit ihnen, **Aufgabenstellung, Architektur und Review bleiben bei mir**.

BERUFSERFAHRUNG

Promova (promova.com) Apr. 2025 – Aug. 2026 · Remote

- Ziel:** das System aus Business-Funnels und öffentlichen SEO-Seiten schnell und für die Teams bedienbar halten; den durchgängigen Web2App-Weg betreuen und die Routine wegautomatisieren.
- Produktarbeit (erstes Jahr — allein).** Migration auf den App Router und Neuaufeilung der Komponenten nach dem neuen Next.js-Modell; Lingui; TanStack Query; Optimierung von Queries und überflüssigen Re-Renders. Funnels, Aufbau von Sections, laufende Aufgaben und Fixes.
- Web2App-Übergang.** Transit-Seite zwischen Anzeige und App Store: das Ergebnis von **AppsFlyer Smart Script (OneLink)** über eine Accessor-Falle auf window abgefangen — die Weiterleitung erfolgte im Moment des Schreibens, ohne Polling. Fallback-Timer und eigener Aufbau der Attribution mit Parameter-Whitelist.
- Analytics.** Amplitude-Events über den gesamten Funnel instrumentiert — vom Anzeigenklick über die Zahlung bis zur Übergabe in die App; ausgewertet, wo genau Nutzer abspringen.
- Zahlungen.** Solidgate und Stripe nach Doku integriert — angebunden und zum Laufen gebracht.
- Häufige UI-Bausteine auf reines CSS umgestellt.** Recherche angestoßen: Modal, Accordion, Slider, Fit-Text — was davon ohne JS funktioniert. Der Ansatz wurde zur Basis eines Open-Source-Pakets (siehe React Patterns).
- Statische Seiten für dynamische Routen.** Ein Skript vorgeschlagen, das sie vor dem Deploy erzeugt. Löste das Turbopack-Problem: eine Seite lädt nur noch ihre eigenen CSS- und JS-Chunks statt aller.
- A/B-Tests vom Next-Layout in den Proxy verlagert** — damit die Seite statisch bleibt.
- Warteschlange für Analytics-Events.** Zwischen Seitenladen und Amplitude-Init konnte ein Nutzer klicken und das Event ging verloren. Eine Queue vorgeschlagen, die Events puffert und nach der Initialisierung nachspielt.
- PR-Review nach Projektzone** (Claude-Agenten). Der Agent erkennt, welchen Teil des Systems eine Änderung betrifft — Performance, Zahlungen, Analytics — und wendet das passende Regelwerk an.
- Influencer-Outreach** — automatisierte Vorbereitung und Versand der Mails statt Handarbeit.
- Yarn** → **pnpm** (Quarantäne für Abhängigkeiten, Isolation, Tempo) · **Redux** → **Zustand** · **Strapi plugins** — Bedarf definiert, Umsetzung geführt, getestet.
- Ergebnis:** vom Abarbeiten von Aufgaben zum Entwerfen und Führen von Lösungen gewechselt. Prozesse aufgebaut, in denen Content-Team und KI-Automatisierung das Maximum selbst erledigen.

Southern Core (Outsourcing, NDA) Sep. 2024 – Feb. 2025 · Remote

- Ziel:** die Projekte auf einen modernen Stack und Produktionsqualität bringen — Architektur, SEO, Stabilität.
- Migration Pages Router** → **App Router**; Architektur auf **Feature-Sliced Design** umgestellt.
- Fehlerbehandlung** — dort ergänzt, wo es keine gab. **UX** — Schwachstellen behoben, verständliche Interface-Logik wiederhergestellt. **API-Request-Batching** — Client↔Server-Roundtrips minimiert.
- Technisches SEO und Barrierefreiheit:** robots.txt, Sitemaps, Meta-Tags, JSON-LD, Canonical, Alt und Title für Bilder, ARIA. **Lokalisierung** — next-intl selbst eingeführt.
- Code-Review für Junior-Entwickler** — Kapselung, Modularität, ausgewogene Wiederverwendung von Komponenten.
- Infrastruktur:** Next.js (ISR) + Directus auf OVH; Deployment über SST + OpenNext auf AWS Lambda; CRON-Automatisierung; A/B über PostHog; CI/CD.
- Ergebnis:** schnelle Seiten (ISR + Batching), geringe Infrastrukturkosten, Content wird über Directus ohne Entwickler gepflegt.

Landee (landee.app) Aug. 2023 – Aug. 2024 · Remote

- Ziel:** ein Baukasten für persönliche Websites (MVP) von Grund auf.
- Architektur.** Code-Struktur und DB-Schema geplant; MVP auf Next.js (App Router), skalierbare FSD-Architektur.
- Block-Editor auf Lexical** (Rich Text: Listen, Tabellen, Code) mit Drag-and-Drop (dnd-kit, GSAP, shadcn/Radix).
- Daten.** MongoDB Atlas über Server Actions + Prisma; vollständiges CRUD; **offline-robuste Synchronisation** — eine Änderungs-Queue mit Rollback. **Eigenes Analytics-System** — Klicks und Impressions, eigene Diagramme.
- Veröffentlichung und Onboarding.** Seiten-Publishing (ISR), Onboarding mit Vorlagen; Auth (Clerk). **Lokalisierung** — next-intl selbst eingeführt (uk/en).
- Serverseite.** Cron-Routen auf Vercel; Turborepo-Monorepo; Anbindung externer APIs (Hintergrundfotos und -videos).
- Frontend und Qualität.** TailwindCSS; GSAP-Animationen (einfache); UI/UX, Fehlerbehandlung; **useEffect minimieren** — ich weiß, wo Reaktivität nötig ist und wo nicht. SEO: Sitemaps, robots.txt, Meta-Tags, JSON-LD.
- Ergebnis:** das MVP des Baukastens ausgeliefert — Editor, Veröffentlichung, Onboarding, Analytics.

EIGENE PROJEKTE · [oleksiimazurenko.dev](#)

- Sortune** — Verwaltung der YouTube-Music-Bibliothek: KI-Agenten über **MCP** klassifizieren Genres und ordnen Playlists nach Saison und Stil. Konzipiert und live gebracht.
- Flint** — **kostenpflichtige macOS-App im Verkauf:** gibt Speicherplatz frei, den Cloud-Dienste belegen (iCloud, Google Drive, OneDrive, Dropbox).
- React Patterns** — Open-Source-Bibliothek auf npm ([@oleksiimazurenko/react-patterns](#)) — entstanden aus Recherche im Produktivbetrieb: Accordion, Dialog, Popover, Slider, Parallax, Reveal, Scroll-Progress, Sticky-Shrink, Fit-Text, Analytics. CSS-first, Server-first, Event-Delegation.
- oleksiimazurenko.dev** — meine eigene Website auf Solid.js: minimales JS, sauberes CSS, schnelles Laden.

EHRENAMT

- Poludne (poludne.com)** — Plattform für ukrainische Literatur: Zitate, Autoren, Werkkatalog. Entwicklung mit **Svelte**, im Team, ehrenamtlich.